

1 **CLAIMS**

2

3 1. A software architecture for a distributed computing system
4 comprising:

5 an application configured to handle requests submitted by remote devices
6 over a network; and

7 an application program interface to present functions used by the
8 application to access network and computing resources of the distributed
9 computing system.

10

11 2. A software architecture as recited in claim 1, wherein the distributed
12 computing system comprises client devices and server devices that handle requests
13 from the client devices, the remote devices comprising at least one client device.

14

15 3. A software architecture as recited in claim 1, wherein the distributed
16 computing system comprises client devices and server devices that handle requests
17 from the client devices, the remote devices comprising at least one server device
18 that is configured as a Web server.

19

20 4. A software architecture as recited in claim 1, wherein the application
21 program interface comprises:

22 a first group of services related to creating Web applications;

23 a second group of services related to constructing client applications;

24 a third group of services related to data and handling XML documents; and

25 a fourth group of services related to base class libraries.

1
2 5. An application program interface embodied on one or more computer
3 readable media, comprising:

- 4 a first group of services related to creating Web applications;
5 a second group of services related to constructing client applications;
6 a third group of services related to data and handling XML documents; and
7 a fourth group of services related to base class libraries.

8
9 6. An application program interface as recited in claim 5, wherein the
10 first group of services comprises:

- 11 first functions that enable construction and use of Web services;
12 second functions that enable temporary caching of frequently used
13 resources;
14 third functions that enable initial configuration;
15 fourth functions that enable creation of controls and Web pages;
16 fifth functions that enable security in Web server applications; and
17 sixth functions that enable access to session state values.

18
19 7. An application program interface as recited in claim 5, wherein the
20 second group of services comprises:

- 21 first functions that enable creation of windowing graphical user interface
22 environments; and
23 second functions that enable graphical functionality.

1 8. An application program interface as recited in claim 5, wherein the
2 third group of services comprises:

3 first functions that enable management of data from multiple data sources;
4 and
5 second functions that enable XML processing.

6

7 9. An application program interface as recited in claim 5, wherein the
8 fourth group of services comprises:

9 first functions that enable definitions of various collections of objects;
10 second functions that enable programmatic access to configuration settings
11 and handling of errors in configuration files;

12 third functions that enable application debugging and code execution
13 tracing;

14 fourth functions that enable customization of data according to cultural
15 related information;

16 fifth functions that enable input/output of data;

17 sixth functions that enable a programming interface to network protocols;

18 seventh functions that enable a managed view of types, methods, and fields;
19 eighth functions that enable creation, storage and management of various
20 culture-specific resources;

21 ninth functions that enable system security and permissions;

22 tenth functions that enable installation and running of services;

23 eleventh functions that enable character encoding;

24 twelfth functions that enable multi-threaded programming; and

25 thirteenth functions that facilitate runtime operations.

1
2 **10.** A network software architecture comprising the application program
3 interface as recited in claim 5.
4

5 **11.** A distributed computer software architecture, comprising:
6 one or more applications configured to be executed on one or more
7 computing devices, the applications handling requests submitted from remote
8 computing devices;
9 a networking platform to support the one or more applications; and
10 an application programming interface to interface the one or more
11 applications with the networking platform.
12

13 **12.** A distributed computer software architecture as recited in claim 11,
14 further comprising a remote application configured to be executed on one of the
15 remote computing devices, the remote application using the application
16 programming interface to access the networking platform.
17

18 **13.** A distributed computer software architecture as recited in claim 11,
19 wherein the application programming interface comprises:
20 a first group of services related to creating Web applications;
21 a second group of services related to constructing client applications;
22 a third group of services related to data and handling XML documents; and
23 a fourth group of services related to base class libraries.
24

1 **14.** A distributed computer software architecture as recited in claim 11,
2 wherein the application programming interface exposes multiple functions
3 comprising:

4 first functions that enable construction and use of Web services;
5 second functions that enable temporary caching of frequently used
6 resources;

7 third functions that enable initial configuration;
8 fourth functions that enable creation of controls and Web pages;
9 fifth functions that enable security in Web server applications; and
10 sixth functions that enable access to session state values.

11
12 **15.** A distributed computer software architecture as recited in claim 11,
13 wherein the application programming interface exposes multiple functions
14 comprising:

15 first functions that enable creation of windowing graphical user interface
16 environments; and
17 second functions that enable graphical functionality.

18
19 **16.** A distributed computer software architecture as recited in claim 11,
20 wherein the application programming interface exposes multiple functions
21 comprising:

22 first functions that enable management of data from multiple data sources;
23 and
24 second functions that enable XML processing.

1 17. A distributed computer software architecture as recited in claim 11,
2 wherein the application programming interface exposes multiple functions
3 comprising:

4 first functions that enable definitions of various collections of objects;
5 second functions that enable programmatic access to configuration settings
6 and handling of errors in configuration files;

7 third functions that enable application debugging and code execution
8 tracing;

9 fourth functions that enable customization of data according to cultural
10 related information;

11 fifth functions that enable input/output of data;

12 sixth functions that enable a programming interface to network protocols;

13 seventh functions that enable a managed view of loaded types, methods,
14 and fields;

15 eighth functions that enable creation, storage and management of various
16 culture-specific resources;

17 ninth functions that enable system security and permissions;

18 tenth functions that enable installation and running of services;

19 eleventh functions that enable character encoding;

20 twelfth functions that enable multi-threaded programming; and

21 thirteenth functions that facilitate runtime operations.

1 18. A computer system including one or more microprocessors and one
2 or more software programs, the one or more software programs utilizing an
3 application program interface to request services from an operating system, the
4 application program interface including separate commands to request services
5 consisting of the following groups of services:

6 A. a first group of services related to creating Web applications:

7 constructing Web services;
8 temporary caching resources;
9 performing initial configuration;
10 creating controls and Web pages;
11 enabling security in Web server applications;
12 accessing session state values;

13 B. a second group of services related to constructing client applications:

14 creating windowing graphical user interface environments;
15 enabling graphical functionality;

16 C. a third group of services related to data and handling XML documents:

17 enabling management of data from multiple data sources;
18 second functions that enable XML processing.

19 D. a fourth group of services related to base class libraries:

20 defining various collections of objects;
21 accessing configuration settings and handling errors in configuration
22 files;
23 debugging and tracing code execution;
24 customizing data according to cultural related information;
25 inputting and outputting of data;

1 enabling a programming interface to network protocols;
2 viewing loaded types, methods, and fields;
3 creating, storing and managing various culture-specific resources;
4 enabling system security and permissions;
5 installing and running services;
6 enabling character encoding;
7 enabling multi-threaded programming; and
8 facilitating runtime operations.

9

10 **19.** A system comprising:

11 means for exposing a first set of functions that enable browser/server
12 communication;

13 means for exposing a second set of functions that enable drawing and
14 construction of client applications;

15 means for exposing a third set of functions that enable connectivity to data
16 sources and XML functionality; and

17 means for exposing a fourth set of functions that enable system and runtime
18 functionality.

19

20 **20.** A system as recited in claim 19, wherein the first set of functions
21 comprises:

22 first functions that enable construction and use of Web services;

23 second functions that enable temporary caching of frequently used
24 resources;

25 third functions that enable initial configuration;

fourth functions that enable creation of controls and Web pages;
fifth functions that enable security in Web server applications; and
sixth functions that enable access to session state values.

21. A system as recited in claim 19, wherein the second set of functions comprises:

first functions that enable creation of windowing graphical user interface environments; and

second functions that enable graphical functionality.

22. A system as recited in claim 19, wherein the third set of functions comprises:

first functions that enable management of data from multiple data sources; and

second functions that enable XML processing.

23. A system as recited in claim 19, wherein the fourth set of functions comprises:

first functions that enable definitions of various collections of objects;

second functions that enable programmatic access to configuration settings and handling of errors in configuration files;

third functions that enable application debugging and code execution tracing;

fourth functions that enable customization of data according to cultural related information;

1 fifth functions that enable input/output of data;

2 sixth functions that enable a programming interface to network protocols;

3 seventh functions that enable a managed view of loaded types, methods,

4 and fields;

5 eighth functions that enable creation, storage and management of various

6 culture-specific resources;

7 ninth functions that enable system security and permissions;

8 tenth functions that enable installation and running of services;

9 eleventh functions that enable character encoding;

10 twelfth functions that enable multi-threaded programming; and

11 thirteenth functions that facilitate runtime operations.

12

13 **24. A method, comprising:**

14 managing network and computing resources for a distributed computing

15 system; and

16 exposing a set of functions that enable developers to access the network and

17 computing resources of the distributed computing system, the set of functions

18 comprising first functions to facilitate browser/server communication, second

19 functions to facilitate construction of client applications, third functions to

20 facilitate connectivity to data sources and XML functionality, and fourth functions

21 to access system and runtime resources.

1 **25.** A method as recited in claim 24, further comprising receiving a
2 request from a remote computing device, the request containing a call to at least
3 one of the first, second, third, and fourth functions.

4

5 **26.** A method, comprising:

6 creating a first namespace with functions that enable browser/server
7 communication;

8 creating a second namespace with functions that enable drawing and
9 construction of client applications;

10 creating a third namespace with functions that enable connectivity to data
11 sources and XML functionality; and

12 creating a fourth namespace with functions that enable system and runtime
13 functionality.

14

15 **27.** A method as recited in claim 26, wherein the first namespace
16 defines classes that facilitate:

17 construction and use of Web services;

18 temporary caching of resources;

19 initial configuration;

20 creation of controls and Web pages;

21 security in Web server applications; and

22 access to session state values.

23

24 **28.** A method as recited in claim 26, wherein the second namespace
25 defines classes that facilitate:

1 creation of windowing graphical user interface environments; and
2 graphical functionality.

3

4 **29.** A method as recited in claim 26, wherein the third namespace
5 defines classes that facilitate:

6 management of data from multiple data sources; and
7 processing of XML documents.

8

9 **30.** A method as recited in claim 26, wherein the fourth namespace
10 defines classes that facilitate:

11 programmatic access to configuration settings and handling of errors in
12 configuration files;

13 application debugging and code execution tracing;

14 customization of data according to cultural related information;

15 inputting and outputting of data;

16 interfacing to network protocols;

17 viewing loaded types, methods, and fields;

18 creation, storage and management of various culture-specific resources;

19 system security and permissions;

20 installation and running of services;

21 character encoding;

22 multi-threaded programming; and

23 runtime operations.

24

25 **31.** A method, comprising:

1 calling one or more first functions to facilitate browser/server
2 communication;

3 calling one or more second functions to facilitate construction of client
4 applications;

5 calling one or more third functions to facilitate connectivity to data sources
6 and XML functionality; and

7 calling one or more fourth functions to access system and runtime
8 resources.

9
10 **32.** A method as recited in claim 36, wherein the first functions
11 comprise functions for construction and use of Web services, temporary caching
12 of resources, initial configuration, creation of controls and pages that will appear
13 as user interfaces, securing Web server applications, and accessing session state
14 values.

15
16 **33.** A method as recited in claim 36, wherein the second functions
17 comprise functions for creation of windowing graphical user interface
18 environments, and graphical functionality.

19
20 **34.** A method as recited in claim 36, wherein the third functions
21 comprise functions for management of data from multiple data sources, and XML
22 processing.

1 **35.** A method as recited in claim 36, wherein the fourth functions
2 comprise functions for programmatic access to configuration settings, application
3 debugging and code execution tracing, customization of text according to cultural
4 related information, synchronous and asynchronous reading from and writing to
5 data streams and files, creation and management of various culture-specific
6 resources, system security and permissions, installation and running of services,
7 character encoding, and multi-threaded programming.

8

9 **36.** A method, comprising:

10 receiving one or more calls to one or more first functions to facilitate
11 browser/server communication;

12 receiving one or more calls to one or more second functions to facilitate
13 construction of client applications;

14 receiving one or more calls to one or more third functions to facilitate
15 connectivity to data sources and XML functionality; and

16 receiving one or more calls to one or more fourth functions to access
17 system and runtime resources.

18

19 **37.** A method as recited in claim 31, wherein the first functions
20 comprise functions for construction and use of Web services, temporary caching
21 of resources, initial configuration, creation of controls and pages that will appear
22 as user interfaces, securing Web server applications, and accessing session state
23 values.

1 **38.** A method as recited in claim 31, wherein the second functions
2 comprise functions for creation of windowing graphical user interface
3 environments, and graphical functionality.

4

5 **39.** A method as recited in claim 31, wherein the third functions
6 comprise functions for management of data from multiple data sources, and XML
7 processing.

8

9 **40.** A method as recited in claim 31, wherein the fourth functions
10 comprise functions for programmatic access to configuration settings, application
11 debugging and code execution tracing, customization of text according to cultural
12 related information, synchronous and asynchronous reading from and writing to
13 data streams and files, creation and management of various culture-specific
14 resources, system security and permissions, installation and running of services,
15 character encoding, and multi-threaded programming.

16

17 **41.** A method for exposing resources using an application program
18 interface, comprising:

19 A. exposing a first group of services related to creating Web applications,
20 including:

- 21 constructing Web services;
22 temporary caching resources;
23 performing initial configuration;
24 creating controls and Web pages;
25 enabling security in Web server applications;

1 accessing session state values;

2 B. exposing a second group of services related to constructing client
3 applications, including:

4 creating windowing graphical user interface environments;

5 enabling graphical functionality;

6 C. exposing a third group of services related to data and handling XML
7 documents, including:

8 enabling management of data from multiple data sources;

9 second functions that enable XML processing.

10 D. exposing a fourth group of services related to base class libraries, including:

11 defining various collections of objects;

12 accessing configuration settings and handling errors in configuration
13 files;

14 debugging and tracing code execution;

15 customizing data according to cultural related information;

16 inputting and outputting of data;

17 enabling a programming interface to network protocols;

18 viewing loaded types, methods, and fields;

19 creating, storing and managing various culture-specific resources;

20 enabling system security and permissions;

21 installing and running services;

22 enabling character encoding;

23 enabling multi-threaded programming; and

24 facilitating runtime operations.

1 **42.** A method of organizing a set of types into a hierarchical namespace
2 comprising:
3 creating a plurality of groups from the set of types, each group containing at
4 least one type that exposes logically related functionality;

5 assigning a name to each group in the plurality; and

6 selecting a top level identifier and prefixing the name of each group with
7 the top level identifier so that the types in each group are referenced by a
8 hierarchical name that includes the selected top level identifier prefixed to the
9 name of the group containing the type.

10
11 **43.** A system comprising:

12 a set of types, each type comprising one of a delegate, an enumeration, an
13 interface, a class, and a structure; and

14 a namespace defined by the set of types to provide access to logically
15 related functionality of a computing system.